Metin İşleme ile PDF'ten Bilgi Çıkarma

Burak İLHAN - Mert TOPRAK
180202058 - 180202074
Bilgisayar Mühendisliği Bölümü
Kocaeli Üniversitesi
180202058@kocaeli.edu.tr - 180202074@kocaeli.edu.tr

***Özet*–Metin işleme tekniklerini kullanarak verilen pdf dosyasından bilgi çıkarma**

Anahtar Kelimeler – Text processing, web

I. PROBLEM TANIMI

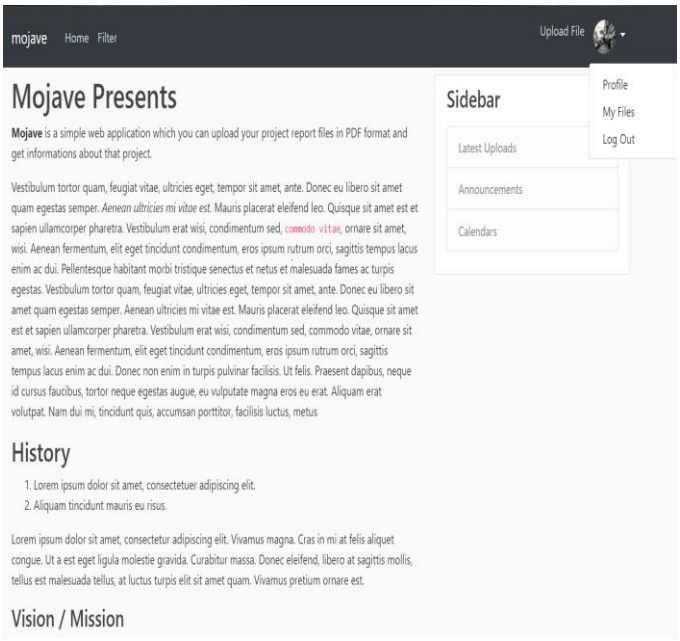
Uygulamamız metin işleme tekniklerini kullanarak verilen pdf formatındaki dosyalardan birtakım bilgileri çıkarmayı amaçlamaktadır.

II. YAPILAN ARAŞTIRMALAR

Projemizi web tabanlı gerçekleştireceğimiz için bir web frameworkü ile çalışmamız gerekiyordu. Öncelikle bu konuda daha önce de geliştirme yaptığımız Django Framework'ünü tercih ettik. Bu framework ve python ile metin işleme teknikleri,pdf okuma gibi konular ile ilgili birkaç tutorial videosu ve web siteden faydalandık ve bunları nasıl kullanmamız gerektiğini öğrendik ve işlemlerimizi gerçekleştirdik. Projemizi VS Code kod editörü kullanarak Python programlama dili ile gerçekleştirdik.

III. TASARIM

Uygulamamızın arayüzü aşağıdaki gibidir.

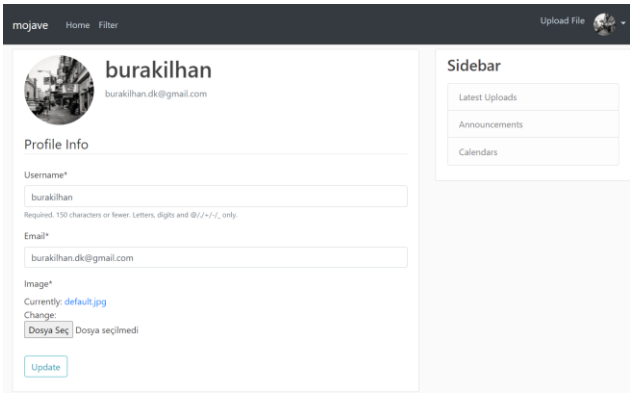


Şekil 1 : Uygulama ana sayfası

IV. GENEL YAPI

Uygulamamız Python programlama dili ve Django Framework'ü kullanılarak VS Code kod editörü üzerinde gerçekleştirilmiştir. Veritabanı olarak Postgresql kullanılmıştır. Arayüz ile başlayacak olursak web sitemize ilk girdiğimizde bizi bir ana sayfa karşılamaktadır. Web uygulamamıza henüz üye olunmadıysa ana sayfadaki navigation barda bulunan register kısmından üyelik işlemlerinin gerçekleştirilmesi gereklidir. Üye olurken kullanıcı adı, şifre ve email bilgileri istenmektedir. Üyelik işlemleri gerçekleştirildikten sonra login kısmından başarılı bir şekilde web uygulamamıza giriş yapabilirsiniz.

Web uygulamamıza giriş yapıldığında sizi Şekil 1'deki gibi bir sayfa karşılayacaktır. Ana sayfada web uygulamamız ile ilgili bilgiler yer almaktadır. Upload Files kısmından uygulamamıza bilgilerinin çıkartılmasını istediğiniz pdf dosyalarınızı yükleyebilirsiniz. Her kullanıcının üye olduktan sonra default bir profil fotoğrafı olur. Bu fotoğrafı ve kullanıcı adı, email gibi bilgilerinizi Profile kısmından değiştirebilirsiniz. My Files kısmında ise uygulamaya yükleme yaptığınız dosyaları görebilir, dosyalardan çıkarılan detaylı bilgilere erişebilirsiniz. Log out kısmı ile birlikte de güvenli bir şekilde uygulamamızdan çıkış yapabilirsiniz.



Şekil 2 : Profile sayfası

Şimdi biraz da bunların arka planda nasıl gerçekleştiğine değinmek istiyorum.
Öncelikle Upload File kısmına tıkladığımızda post/create/ urlsine yönlendiririliriz. Bu url'de bizi dosyamızı yükleyebileceğimiz bir form beklemektedir. Dosya seç butonuna basarak bilgisayarımızdan yüklemek istediğimiz pdf dosyasını seçeriz. Sonrasında ise Upload butonuna basarak dosyamızı web uygulamasına yükleyebiliriz. Upload butonuna basıldığında uygulamamızın backend kısmında dosya var mı, varsa geçerli format mı gibi controller yapılır. Eğer bu konularda bir sorun yoksa pdf, kullanıcı bilgisi ile birlikte veritabanına kaydedilir. Daha sonrasında ise yüklenen pdf'ten gerekli bilgileri çıkarmak için read_pdf fonksiyonu çağırılır. Bu fonksiyon, yüklenen dosyanın id numarasını ve yüklendiği pathi parametre olarak alır. Bu parametreleri yollarak read_pdf fonksiyonunu çağırırız.

```
def read_pdf(pathname,file_id):
    all_pdf = []
    with fitz.open(pathname) as doc:
        text = ""
        for page in doc:
            text = page.get_text()
            all_pdf.append(text)

    all_pdf[0] = all_pdf[0].replace('\n','')
    all_pdf[0] = all_pdf[0].replace("..","")
    all_pdf[0] = all_pdf[0].split("  ")
    str_list = list(filter(None, all_pdf[0]))
    for i in range (0,len(str_list)):
        str_list[i] = str_list[i].strip()
    lesson = str_list[3]
    title = str_list[4]
    student_name = str_list[5]

    supervisor = ""
    judges = []

    for i in range(len(str_list)):
        if "Dan□şman" in str_list[i]:
            str_list[i] = str_list[i].replace("Dan□şman","")
            name = str_list[i].split(",")
            supervisor = name[0].strip()
        if "Jüri" in str_list[i]:
            str_list[i] = str_list[i].replace("Jüri Üyesi","")
            name = str_list[i].split(",")
            judges.append(name[0].strip())
```

Şekil 3 : read_pdf fonksiyonu

Bu fonksiyon içerisinde öncelikle parametre olarak pathi verilen pdf açılır, daha sonrasında ise bir python kütüphanesi olan fitz kütüphanesi ile pdf dosyasından okuma işlemleri gerçekleştirilir. Okunan her bir sayfa, ayrı ayrı olmak üzere all_pdf isimli listeye eklenir. Bu işlemler yapıldıktan sonra okunan sayfalar üzerinde boşlukları kaldırma, gereksiz noktalamaları kaldırma gibi işlemler yapılır. Sonrasında ise her bir sayfa satırlara ayrılır. Proje dosyalarının formatı gereği bazı bilgilerin nerelerde olacağı bellidir. Örneğin öğrenci ismi, proje isminin altında olur. Bu klasik formattan yararlanarak pdf dosyasının ilk sayfasından öğrenci ismi, proje ismi ve ders bilgileri çıkarılır ve değişkenlere atanır.

Sonrasında ise projenin danışman ve jüri bilgilerinin çıkarılma işlemi başlanır. Bu bilgilerin hangi sayfalarda olacağı kesin değildir, dolayısıyla bütün sayfalar tek tek kontrol edilir ve içerisinde "Danışman" ve "Jüri" metini olan sayfanın sayısı bir değişkende tutulur. Daha sonrasında ise bu sayfa üzerinde işlemler yapılır ve Danışman metninin bulunduğu kısımdaki akademisyen, danışman olarak alınır ve bir değişkene kaydedilir. Sonrasında Jüri kısmındaki akademisyenlerin bulunduğu kısım öncelikle bir string içerisinde çıkarılır, daha sonra bu stringdeki akademisyen isimleri bir listeye atılarak jüri bilgileri çıkarılmış olunur.

```
matches = list(datefinder.find_dates(str_list[-1]))

if len(matches) > 0:
    date = matches[0]
else:
    print('No dates found')

if date.day > 1 and date.day < 9:
    delivery_date = str(date.year - 1) + "-" + str(date.year) + " Bahar"
else:
    delivery_date = str(date.year) + "-" + str(date.year+1) + " Güz"
```

Şekil 4 : Dönem bilgisi

Sonrasında ise projenin hangi dönemde yapıldığı bilgisi aranır. Teslim tarihleri genellikle raporların ilk sayfasında yer alır. Bundan dolayı öncelikle ilk sayfadaki metinler kontrol edilir ve içerisinde herhangi bir tarih varsa, bu tarih bir değişkene atılır. İlk sayfada herhangi bir tarih olmaması durumunda aynı kontrol diğer sayfalar için de yapılır ve bulunan tarih, bir değişkende tutulur. Sonrasında ise bu tarih üzerinde işlem yapılır. Tarihteki ay Şubat ile Eylül arasında ise bahar dönemi, diğer durumda ise güz dönemidir. Teslim tarihi bahar dönemine denk geliyorsa dönem bilgisi "teslim tarihindeki yılın bir önceki yılı – teslim tarihinin yılı Bahar" şeklinde tutulur. Örneği 01.06.2020 teslim tarihli bir projenin dönemi 2019-2020 Bahar dönemi olarak kaydedilir. Aksi durumda ise "teslim tarihindeki yıl – teslim

tarihindeki yılın sonraki yılı Güz" şeklinde tutulur. Örneğin 01.12.2020 teslim tarihli bir projenin dönemi "2020-2021 Güz" dönemi olarak kaydedilir.

Sonrasında ise öğrenci numarası ve öğrencinin eğitim türü bilgilerinin çıkarılması işlemleri başlar. Bunun için pdf'te öğrenci numarasının olduğu pdf sayfası tespit edilir. Pyton'ın re isimli kütüphanesi sayesinde bu sayfa içerisindeki metinden sayıların çıkarma işlemi gerçekleştirilir ve öğrenci numarası olarak değişkende tutulur. Daha sonrasında ise öğrencinin eğitim türü bilgisi bulunmaya çalışılır. Uygulamamıza yükleyeceğimiz pdf dosyaları Kocaeli Üniversitesi tez raporu formatında olmaktadır. Dolayısıyla eğitim türü bilgisi de Kocaeli Üniversitesi'ndeki gibi bulunmaktadır. Bu da öğrenci numarasından yola çıkılarak bulunabilir. Kocaeli Üniversitesinde örgün eğitim olanların öğrenci numarasının 5. Hanesi 1, ikinci öğretim olanların ise 2'dir. Buradan yola çıkılarak çıkarılan öğrenci numarasının 5. Hanesi kontrol edilir. Eğer 1 ise eğitim türü bilgisi örgün, 2 ise gece olarak bir değişkende tutulur.

```
all_pdf[2] = all_pdf[2].replace('\n','')
all_pdf[2] = all_pdf[2].split("  ")
str_list3 = list(filter(None, all_pdf[2]))
for i in range (0,len(str_list3)):
    str_list3[i] = str_list3[i].strip()

student_no = re.findall(r'\d+', str_list3[-1
student_no = str(student_no[0])

type_of_edu = ""

if(student_no[5] == "2"):
    type_of_edu = "Gece"
else:
    type_of_edu = "Örgün"

student_no = int(student_no)
```

Şekil 5 : Öğrenci No ve Eğitim Türü Bilgisi

Son olarak da raporun özet ve anahtar kelimeler bilgisi çıkarılır. Bunun için bütün pdf'teki sayfaların tek tek kontrolü yapılır, içerisinde "Özet" ve "Anahtar" kelimeleri geçen sayfanın index bilgisi bir değişkende tutulur. Daha sonra bu sayfa önceki sayfalarda da yapıldığı gibi birkaç düzenleme işlemi yapılır. Sonrasında ise "Özet" yazısından "Anahtar" yazısına kadar olan metin, Özet kısmı olarak çıkarılır ve bir değişkende tutulur. Sonrasında ise "Anahtar" kelimesinden sonra cümle bitene kadar olan kelimeler, bir liste değişkeninde tutulur. Bütün bu bilgiler elde edildikten parametre olarak aldığımız id bilgisi ile, veritabanından yüklenen pdf'i çağırırız ve elde edilen bilgileri de ekleyerek pdf dosyasını güncelleyerek veritabanına kaydederiz.

```
obj = Pdf.objects.get(id=file_id)
obj.title=title
obj.student=student_name
obj.lesson=lesson
obj.season=delivery_date
obj.keywords=keywords_list
obj.judges=judges
obj.supervisor=supervisor
obj.summary=summary
obj.student_no = student_no
obj.type_of_edu = type_of_edu
obj.save()
```

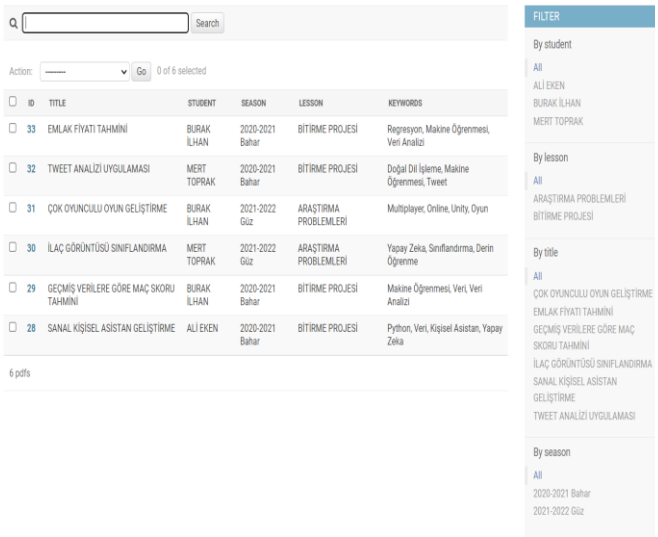
Şekil 6: Pdf dosyasının güncellenmesi

Bu işlemler gerçekleştikten sonra dosyayı yükleme işlemimiz başarıyla gerçekleşir ve uygulama ana sayfasında yönlendiririliriz. Buradan My Files kısmından yüklediğiniz pdf dosyalarını görebilir, detaylı bilgilerine erişebilirsiniz.

```
def upload_file(request):
    if request.method == 'POST':
        file = request.FILES['fupload']
        pdf = Pdf(file=file, author=request.user)
        pdf.save()
        read_pdf(pdf.file.path,pdf.id)
        return redirect('/')
    return render(request, 'blog/upload.html')
```

Şekil 7: Pdf dosyasının kaydedilmesi

Son olarak da uygulamamızın yönetim paneli kısmından bahsetmekte fayda var. Uygulamamızda admin tarafından erişilebilen ve bütün kayıtlı kullanıcı ve onların yükledikleri pdf dosyalarının admin tarafından görüntülenebildiği ve manipüle edilebildiği bir yönetici paneli bulunmaktadır. Bu yönetici panelinde admin kullanıcı ekleme, silme, güncelleme gibi işlemleri gerçekleştirebilmektedir. Bunun yanı sıra, yüklenen pdf'leri ve bu pdf'lerden çıkarılan ve veritabanına kaydedilen bilgileri görüntüleyebilmektedir. Ayrıca bu pdf dosyaları üzerinde sorgular da gerçekleştirebilmektedir. Örneğin belirli bir öğrencinin, kullanıcının yüklediği projelere ulaşabilir, bunun dışında belirli bir kullanıcının belirli bir dönemde yüklediği projeler gibi sorgular da gerçekleştirebilmektedir.



Şekil 8: Yönetici Paneli

V. SONUÇ

Bu projeyi geliştirirken bir pdf dosyasını nasıl okuyabileceğimizi ve okuduğumuz dosyadan metin işleme teknikleri ile nasıl bilgi çıkarabileceğimizi öğrenmiş olduk. Ayrıca bunu bir web uygulama olarak yaparak aynı zamanda web programlama konusunda da yeni bilgiler öğrendik ve bu konularda kendimize yeni şeyler kattık.

VI. REFERANSLAR

1) https://stackoverflow.com/questions/11754877/troubleshooting-related-field-has-invalid-lookup-icontains

2) https://django-filter.readthedocs.io/en/stable/

3) https://docs.djangoproject.com/en/4.0/topics/db/queries/

4) https://www.youtube.com/watch?v=UmljXZIypDc&list=PL-osiE80TeTtoQCKZ03TU5fNfx2UY6U4p

5) https://www.youtube.com/watch?v=G-Rct7Na0UQ&t=94s

6) https://www.tutorialspoint.com/python_text_processing/index.htm

7) https://stackoverflow.com/questions/45795089/how-can-i-read-pdf-in-python

8) https://github.com/pymupdf/PyMuPDF